Barcelona Supercomputing Center
Centro Nacional de Supercomputación

# R user meeting

An-Chi Ho, Eva Rifà, Victòria Agudetse

contributor: Alba Llabrés-Brustenga

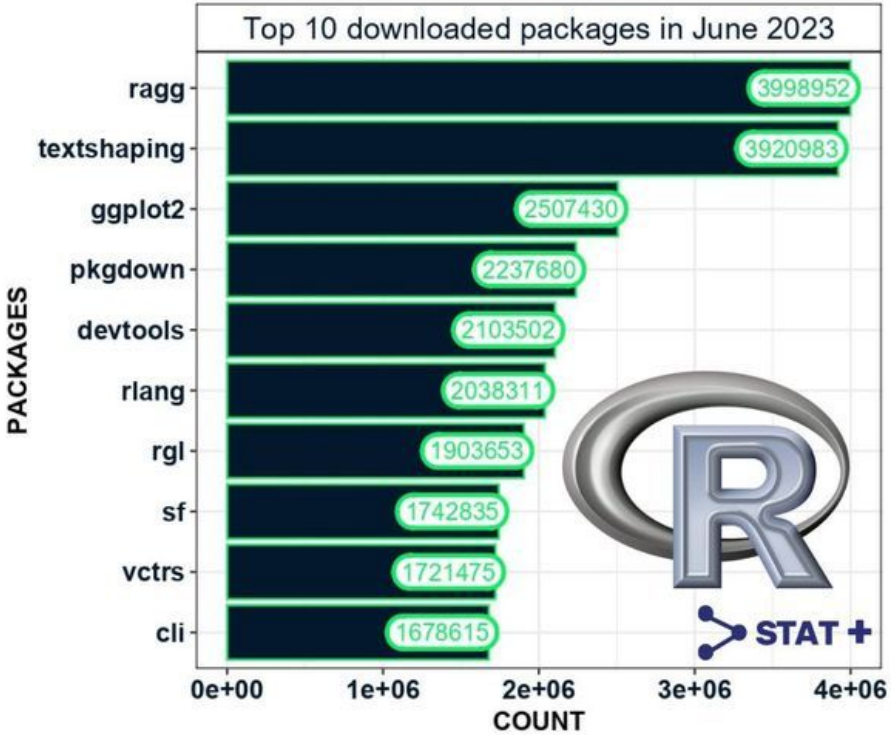
06/07/2023

# Agenda

1. Ice-breaker: Most downloaded R package in June 2023
2. News
   - General R
   - s2dv
   - startR
   - CSTools
   - CSIndicators
   - SUNSET
3. Presentation: SPEI [Alba]
4. Q&A

# Ice-breaker

# Most downloaded R package in June 2023



# General R

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

# Corrected point intersection shp_mask

- The function **shp_file** returned error when the intersection of the coordinates and the shapefile specific region was a point.

**Example**

```
shp.file <- paste0('/esarchive/shapefiles/NUTS3/NUTS_RG_60M_2021_4326.shp/',
                   'NUTS_RG_60M_2021_4326.shp')
ref.grid <- paste0('/esarchive/exp/ecmwf/s2s-monthly_ensfor/weekly_mean/',
                   'tas_f6h/tas_20191212.nc')
NUTS.id <- paste0("FI1D", c(1:3, 5, 7:9))
mask1 <- shp_mask(shp.file, ref.grid, reg.ids = NUTS.id)
```

```
Error in if (nrow(tmp_coords) != 0) { : argument is of length zero
```

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

status: In branch develop-shp_mask
Check issue: https://earth.bsc.es/gitlab/es/s2dv/-/issues/75

# A new package for plotting functions?

- Put all the plotting functions in s2dv and CSTools into a new package
- Advantages:
  - Easy to maintain
  - Easy to use functions together
  - Avoid complex dependencies of one package

# s2dv

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

# Load(): parameter “grid”

Supplement of documentation of parameter `**grid**`:

```
#'  Note that the auto-detected grid type is not guarenteed to be correct, and
#'  it won't be correct if the netCDF file doesn't contain global domain.
#'  Please check the warning carefully to ensure the detected grid type is
#'  expected, or assign this parameter even regridding is not needed.
```

Suggestions:

(1) Use **startR** to load data

(2) If use Load(), use parameter “grid” even if the data don’t need to be regridded.

→ Avoid wrong output if the grid type is not recognized by Load()

→ Necessary for regional netCDF files

# Combine PlotStereoMap by self-defined layout

To combine PlotStereoMap() outputs in one plot,

- Share common color bar ⟶ PlotLayout()
- Use different color bars ⟶ Define layout() first, then plot figures and color bars separately

Explanation: When colorbar is required (`drawleg = T`), PlotStereoMap() uses `layout()` to define the layout of figure and color bar. Hence, the previous defined `layout()` will be overwritten. The trick is to plot figure and color bar separately.

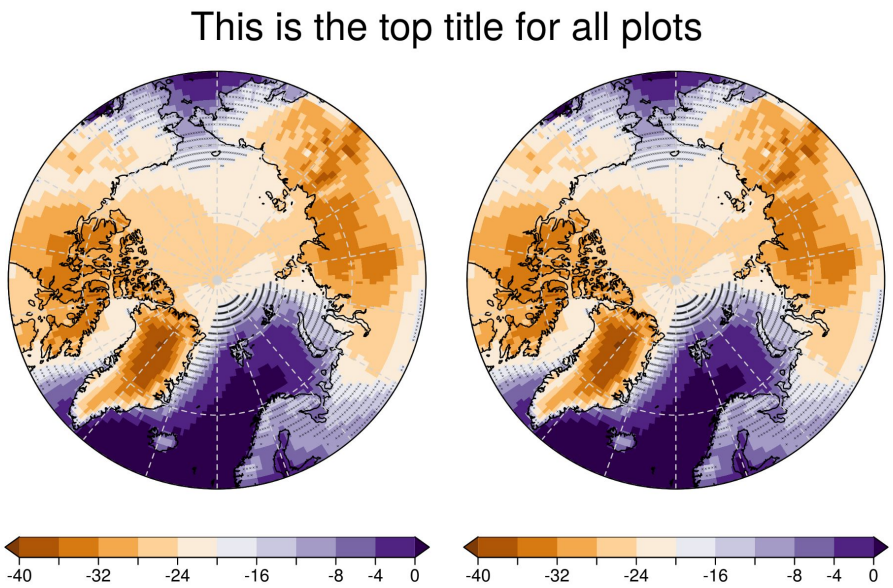
See full example:
https://earth.bsc.es/gitlab/aho/aho-testtest/-/blob/master/s2dv/script_PlotStereoMap.R

# Combine PlotStereoMap by self-defined layout

```
#------- Manually create layout -------

png("fig1.png", width = 22, height = 14, unit = 'cm', res = 300)

layout.matrix <- matrix(c(1, 2, 3, 4), nrow = 2, ncol = 2)

layout(mat = layout.matrix, heights = c(lcm(10), lcm(2)), width = c(lcm(10), lcm(10)))

#layout.show(4)

#-------- Plot fig 1 -------

PlotStereoMap(data1, lon = lon, lat = lat, brks = brks, dots = mask,
              filled.continent = F, drawleg = F)

#------- Plot colorBar for Fig 1 -------

ColorBar(brks = brks, vertical = F, var_limits = range(data, na.rm = T),
         extra_margin = c(0.5, 0, 0.5, 0))

#-------- Plot fig 2 -------

PlotStereoMap(data2, lon = lon, lat = lat, brks = brks, dots = mask,
              filled.continent = F, drawleg = F)
```

# Combine PlotStereoMap by self-defined layout

```
#------- Plot colorBar for Fig 2 -------

ColorBar(brks = brks, vertical = F, var_limits = range(data, na.rm = T),
         extra_margin = c(0.5, 0, 0.5, 0))

#------- Top title -------

mtext("This is the top title for all plots", side = 3, line = -2, outer = TRUE, cex = 2)

dev.off()
```



# startR

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

# Retrieve variable metadata when `retrieve = F`

Start() now returns variable's (e.g., tas) metadata when retrieve = F. It uses NcOpen() to open the file and get variable attributes, which is different from retrieve = T, so the metadata have slight differences.

Examples: https://earth.bsc.es/gitlab/es/startR/-/issues/118#note_219229

status: in master

```
 names(attr(dataF, 'Variables')$common)
```

```
[before]
#[1] "lat" "lon"

[now]
#[1] "lat" "lon" "tas" "clt"
```

# Loading memory limitation on Nord3v2

It seems that startR has a 16G data size limitation when retrieving data, no matter how much RAM requested. We're not sure if it is due to startR or our infrastructure. You can follow this issue here: https://earth.bsc.es/gitlab/es/startR/-/issues/179

Data is too big? → Use Compute()

# Get margin dimemsion indices when using Compute()

Goal: To know the margin dimension indices in the self-defined function.

Imagine you have:

- Data array `[dat = 1, var = 2, sdate = 8, ensemble = 25, time = 3, lat = 61, lon = 121]`
- The two variables are ‘psl’ and ‘tas’.
- You want 4 chunks, `[var = 2, sdate = 2]`

You want to change tas unit from K to degC in the function. So, the deduction is only applied in the two chunks with `var = 2`.

Use case:

https://earth.bsc.es/gitlab/es/startR/-/blob/master/inst/doc/usecase/ex2_14_margin_dim_indices.R

# CSTools

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

# New function CST_Start ?

- This function could be used to call Start obtaining as a result an object of class 's2dv_cube'. Similarly to CST_Load.

```
CST_Start <- function(...) {
  exp <- Start(...)
  result <- as.s2dv_cube(exp)
  result
}
```



status: Not developed yet

# CSIndicators

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

# New element in ‘s2dv_cube’: `time_bounds`

- Added in **‘s2dv_cube’ output** of functions that aggregate time: **CST_PeriodMean**, **CST_PeriodAccumulation**, **CST_AccumulationExceedingThreshold**, **CST_TotalSpellTimeExceedingThreshold**, **CST_TotalTimeExceedingThreshold**
- The element ‘`time_bounds`’ contain the first and end Dates of the period aggregation with the same dimensions as Dates.

**Example**

```
res <- CST_PeriodMean(data = CSTools::lonlat_prec, start = list(10, 03), end = list(20, 03))
```

```
$ data : ...
$ coords: ...
$ attrs : ...
..$ Variable : ...
..$ Dates : POSIXct[1:3],   "2011-03-10 UTC" "2012-03-10 UTC" "2013-03-10 UTC"
..$ time_bounds :
.. ..$ start: POSIXct[1:3], "2011-03-10 UTC" "2012-03-10 UTC" "2013-03-10 UTC"
.. ..$ end  : POSIXct[1:3], "2011-03-20 UTC" "2012-03-20 UTC" "2013-03-20 UTC"
```

**Original dates:**

```
> dim(lonlat_prec$attrs$Dates)
ftime sdate
   31     3
```

```
> dim(res$attrs$Dates)
sdate
    3
```

```
> dim(res$attrs$time_bounds$start)
sdate
    3
```

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

status: In branch [develop-Dates_periodmean]
Check issue: https://earth.bsc.es/gitlab/es/csindicators/-/issues/24

# Improved use of `start` and end

- Explicitly mentioned in documentation that Dates must have dimensions.
- Parameters ‘start’ and ‘end’ will not be used to select period if Dates dimensions are NULL.

```
if (!is.null(dim(dates))) {
  data <- SelectPeriodOnData(data = data, dates = dates, start = start,
                             end = end, time_dim = time_dim, ncores = ncores)
} else {
  warning("Parameter 'dates' must have named dimensions if 'start' and ",
          "'end' are not NULL. All data will be used.")
}
```

status: In branch develop-Dates_periodmean
Check issue: https://earth.bsc.es/gitlab/es/csindicators/-/issues/24

# Default value of time_dim

- Change all default values of **time_dim** to ‘time’.
- Now, CST functions have by default **time_dim =** ‘ftime’ while others have **time_dim =**‘time’.

**Example**

```
CST_PeriodMean <- function(data, start = NULL, end = NULL,
                           time_dim = 'ftime', na.rm = FALSE,
                           ncores = NULL)

PeriodMean <- function(data, dates = NULL, start = NULL, end = NULL,
                       time_dim = 'time', na.rm = FALSE, ncores = NULL)
```



status: Not developed yet; included in next release

# SUNSET

# New module: Indices

A new module **Indices**, created by Núria Pérez-Zanón, is now available in the master branch. It can perform the computation, save the values, and create spatial pattern and time series plots for the following indices:

- **NAO** (variables: psl, z500)
- **Niño1+2, Niño3, Niño3.4 and Niño4** (variables: tas, SST)

It is compatible with all other modules in the tool. For now, this module only works with the **hindcast** and **observations**. The **forecast** development is pending.

status: In master

# New module: Indices

- NAO recipe options:

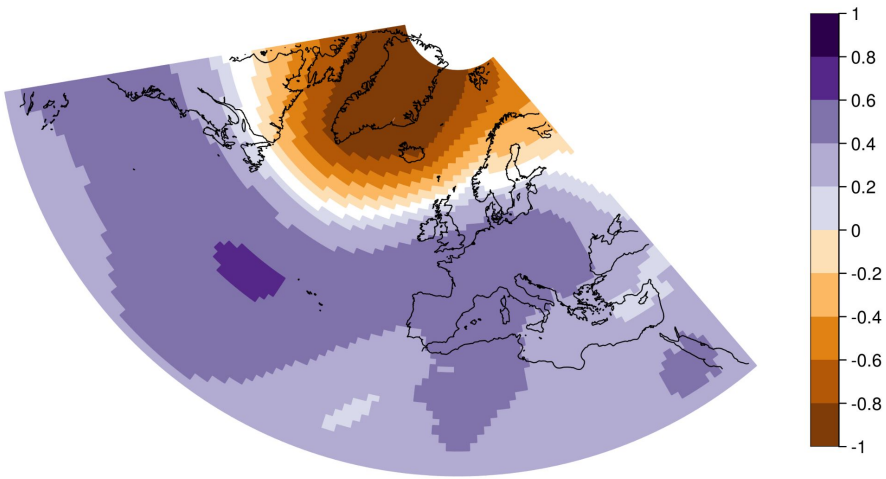
```
Indices:

  NAO: {obsproj: yes/no, save: 'all'/'none', plot_ts: yes, plot_sp: yes/no}
```

- El Niño recipe options:

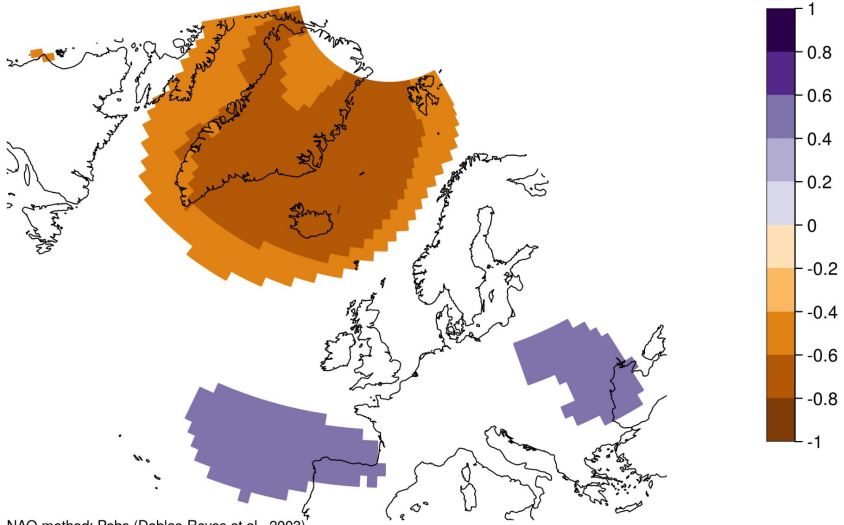
```
Indices:

  Nino1+2: {save: 'all'/'none', plot_ts: yes/no, plot_sp: yes/no}

  Nino3:   {save: 'all'/'none', plot_ts: yes/no, plot_sp: yes/no}

  Nino3.4: {save: 'all'/'none', plot_ts: yes/no, plot_sp: yes/no}

  Nino4:   {save: 'all'/'none', plot_ts: yes/no, plot_sp: yes/no}
```

Example scripts and recipes available here: SUNSET example scripts

# NAO plots

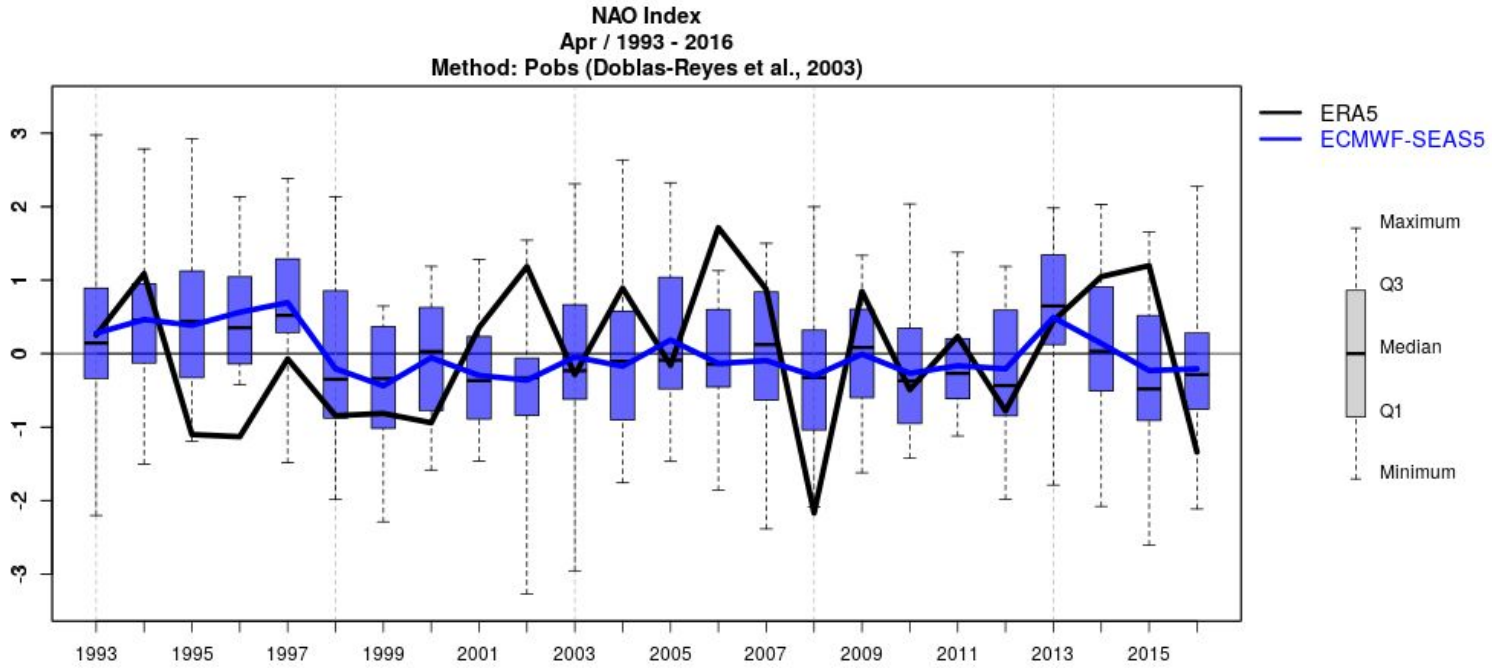



NAO method: Pobs (Doblas-Reyes et al., 2003)
Correlation all members
Nominal start date: 1st of March
Forecast month: 02
Pearson correlation; alpha = 0.05



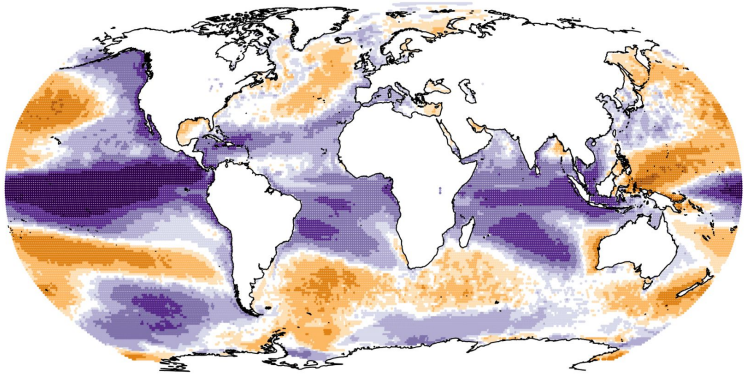

NAO method: Pobs (Doblas-Reyes et al., 2003)
Nominal start date: 1st of March
Forecast month: 02
Pearson correlation; alpha = 0.05

# NAO plots

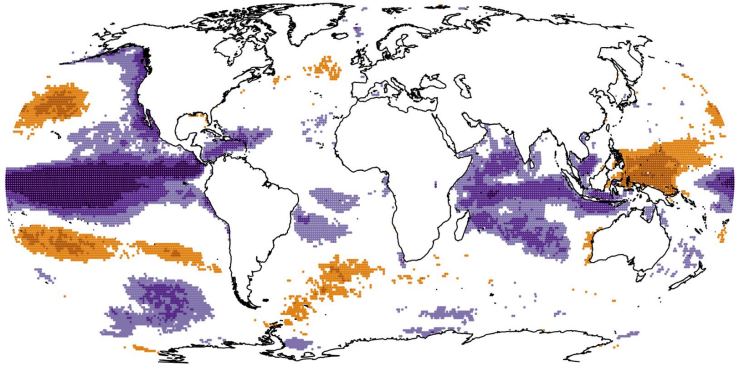



NAO method: Pobs (Doblas-Reyes et al., 2003)
Nominal start date: 1st of March
Forecast month: 02

# Niño plots

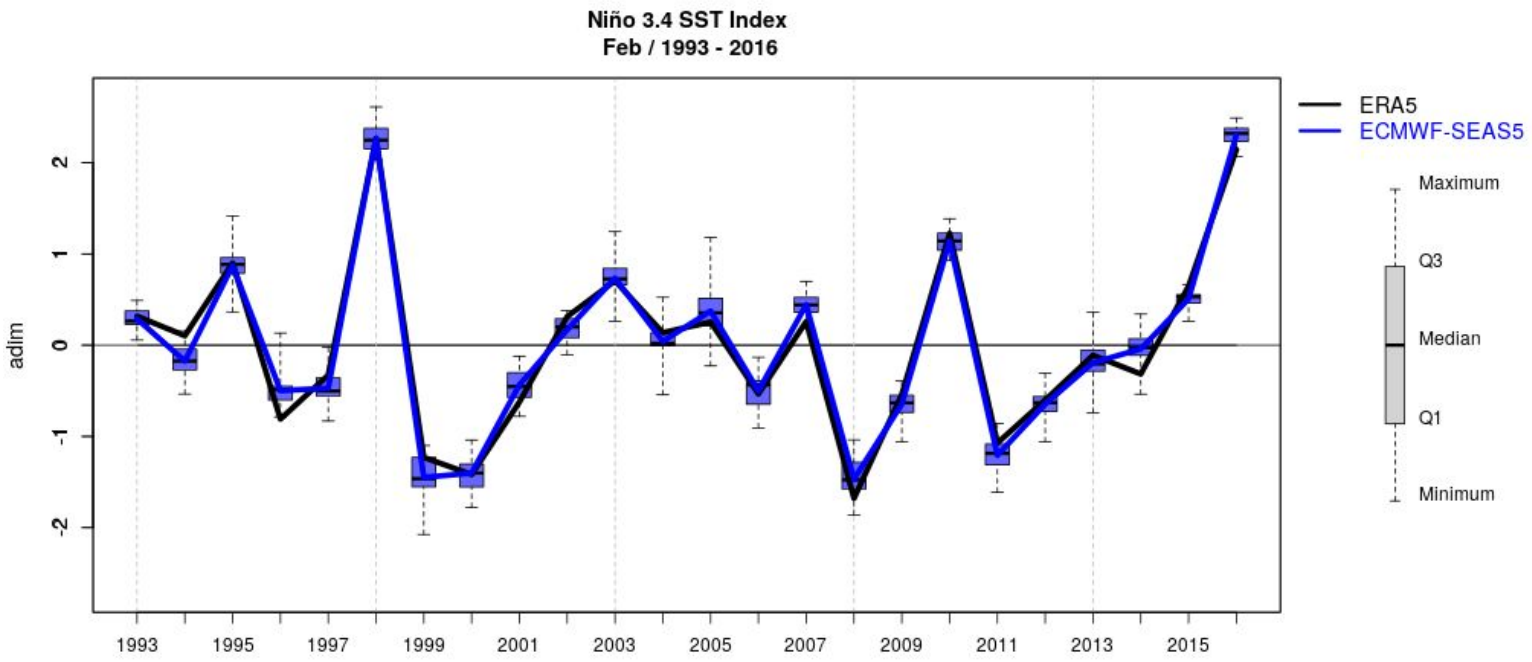



Individual members
Nominal start date: 1st of January
Forecast month: 02
Pearson correlation ; alpha = 0.05




Nominal start date: 1st of January
Forecast month: 02
Pearson correlation ; alpha = 0.05

# Niño plots




Nominal start date: 1st of January
Forecast month: 02

# User presentation

# SPEI: Standardised Precipitation-Evapotranspiration Index

more information:
https://spei.csic.es/home.html

(positive values)
wetness

dryness
(negative values)

Indices to monitor **drought**:

→ Palmer Drought Severity Index (PDSI)

- ✅ It is based on water balance equation (supply vs demand)
- ✅ It considers: precipitation, moisture supply, runoff and evaporation at surface
- ❌ It is defined at a fixed temporal scale (between 9 and 12 months)

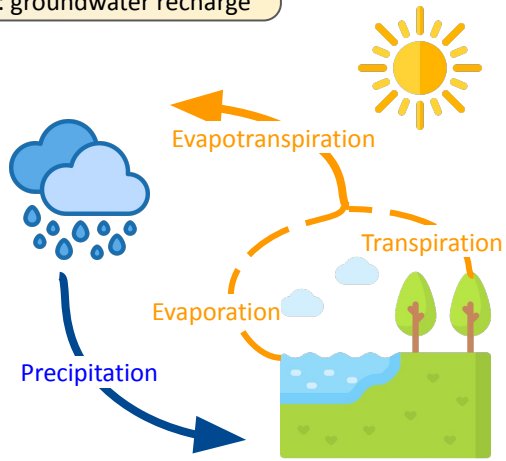
→ Standardised Precipitation Index (SPI)

- ✅ It can be calculated at **different time scales** →
  - **Meteorological droughts** (1 to 3 months): immediate impacts
  - **Agricultural droughts** (6 to 12 months): reservoir storage
  - **Hydrological droughts** (12 to 24 months): groundwater recharge
- ❌ It neglects the importance of temperature and potential evapotranspiration

→ Standardised Precipitation-Evapotranspiration Index (SPEI)

- ✅ It can be calculated at different time scales
- ✅ It considers the effect of **potential evapotranspiration (PET)** on drought severity

*under "ideal" circumstances (assuming a water supply, no resistance to the flow of water, …)*
*→ not an actual measurement*




Climate Water Balance (CWB)

# SPEI: Standardised Precipitation-Evapotranspiration Index

## Step 1: Potential Evapotranspiration **(PET)**

The current methods of PET estimation available in CST_PeriodSPEI are the following (temperature-based empirical approximations):

- **Hargreaves**: minimum temperature, maximum temperature, latitude
- Hargreaves modified: minimum temperature, maximum temperature, precipitation, latitude
- Thornthwaite: temperature (monthly mean), latitude

Latitude is used to estimate the solar radiation (depending on the time of the year).

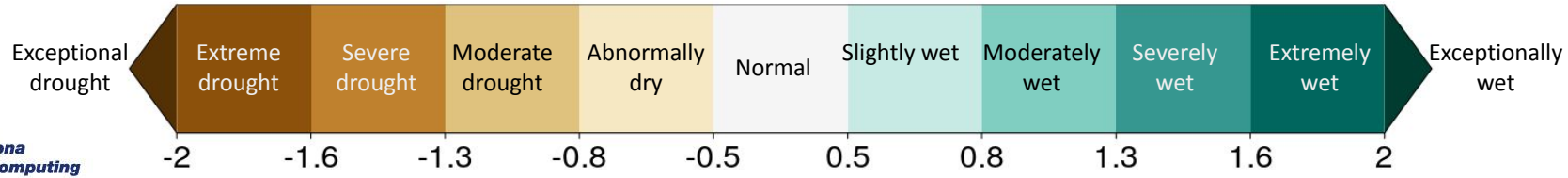
However, other methods exist (physical approaches like the Penman Monteith equation) → the function allows the user to provide PET directly as a parameter and, in that case, the step of PET estimation is skipped.

## Step 2: Accumulation

Decision about the time scale: e.g. **SPEI1**, **SPEI3**, **SPEI6**

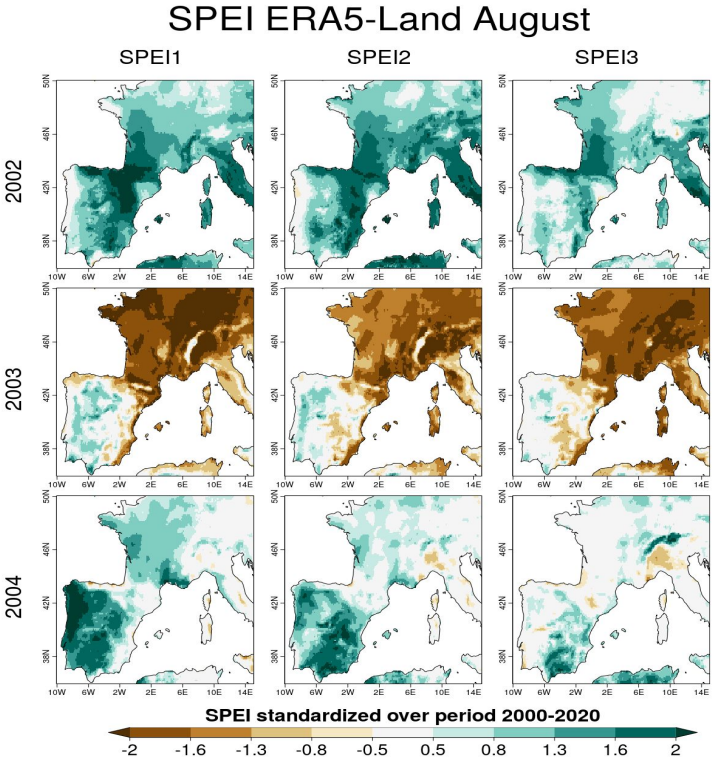
## Step 3: Standardisation

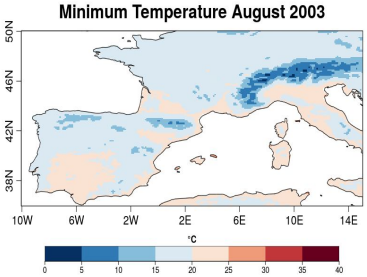
It is an optional step, but necessary for comparison of the drought index across time and space. **Long time series** are needed for the standardisation. The **forecast** will be standardised with the information from the **hindcast**.

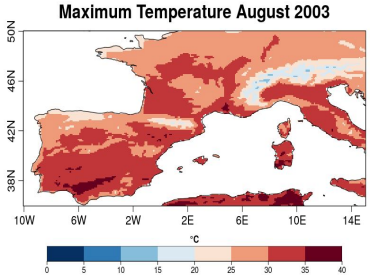


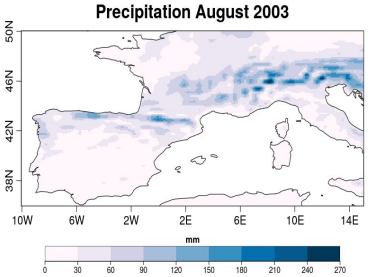

Classification adapted from **Marco Turco *et al* 2017 *Environ. Res. Lett.* 12 084006** and following MeteoSwiss

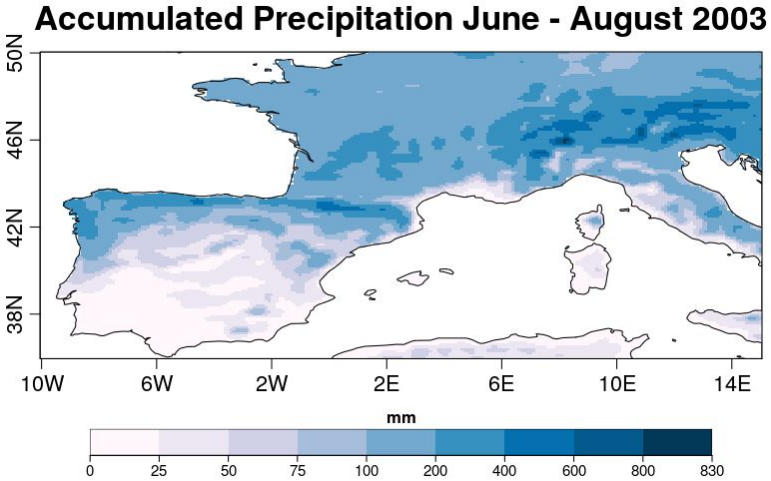
# SPEI: Standardised Precipitation-Evapotranspiration Index



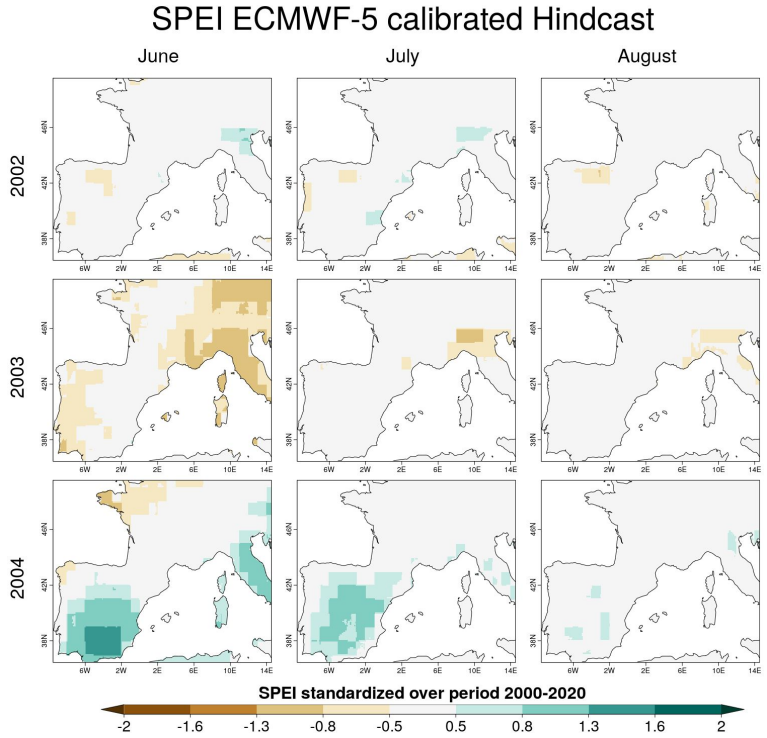



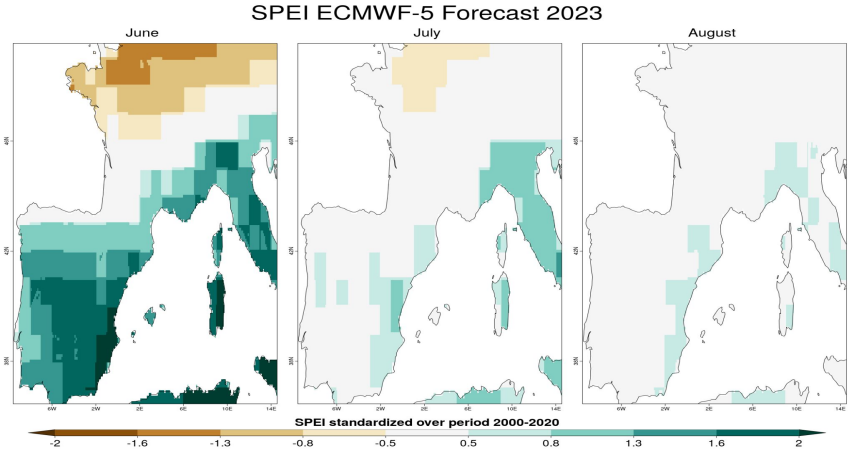

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

# SPEI: Standardised Precipitation-Evapotranspiration Index




1. Calibration of the accumulation *(in this case monthly values)* of the difference prlr - PET
2. Standardization of the calibrated accumulation to obtain SPEI1

Script (running time in WS:1.93 hours)



# Thanks for joining

Next meeting: 7th September, 12h